APPENDIX - SOURCE CODE

```
/* Query 1:
    5
              select x, b
                from S, T
               where y = a
              Plan:
   10
                       BNO -- [ s.#0, t. #1 ]
                         -- [ s.#1 = t.#0 ]
                     NLJ
   15
                     / \
              TA S s
                         TA T t
             */
   20
        #if O
             /* BNO */
             thread[0].opc = eBNO;
thread[0].typ = eT_streamof(c10vc10);
   25
             thread[0].obj = (ePtr)objoffset(bno, heap);
                           = (ePtr)3;
             bno->in
             bno->fn
                           = (ePtr)1;
                          = 2;
             bno->fns
                          = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
a.
   30
             /* s.#0 */
Ħ
             thread[1].opc = eVARATT;
thread[1].typ = eT_setlen(eT_char, 10);
             thread[1].obj = (ePtr)objoffset(varatt1, heap);
             varatt1->tvar = (ePtr)9;
   35
             varatt1->col = 0;
N
             /* t.#1 */
             thread[2].opc = eVARATT;
             thread[2].typ = eT_setlen(eT_varchar, 10);
   40
             thread[2].obj = (ePtr)objoffset(varatt2, heap);
             varatt2->tvar = (ePtr)10;
             varatt2->col = 1;
   45
             /* NLJ */
             thread[3].opc = eNLJ;
             thread[3].typ = 0;
             thread[3].obj = (ePtr)objoffset(nlj, heap);
                          = (ePtr)7;
             nlj->outer
   50
                          = (ePtr)8;
             nlj->inner
             nlj->pred
                           = (ePtr)4;
             /* == */
             thread[4].opc = eEQ_II;
   55
             thread[4].typ = eT_bool;
             thread[4].obj = (ePtr)objoffset(eq_ii, heap);
```

```
= (ePtr)5;
            eq_ii->x1
             eq_ii->x2
                          = (ePtr)6;
             /* s.#1 */
   5
            thread[5].opc = eVARATT;
             thread[5].typ = eT_int;
             thread[5].obj = (ePtr)objoffset(varatt3, heap);
            varatt3->tvar = (ePtr)9;
            varatt3->col = 1;
   10
             /* t.#0 */
             thread[6].opc = eVARATT;
             thread[6].typ = eT_int;
             thread[6].obj = (ePtr)objoffset(varatt4, heap);
   15
            varatt4->tvar = (ePtr)10;
            varatt4->col = 0;
             /* TA S s */
             thread[7].opc = eTA;
             thread[7].typ = eT_streamof(c10intsml);
   20
             thread[7].obj = (ePtr)objoffset(ta1, heap);
             ta1->tvar
                          = (ePtr)9;
             ta1->pred
                           = (ePtr)0;
ta1->tmd
                           = t->pTMD;
   25
             ta1->openp
                          = 0;
             /* TA T t */
             thread[8].opc = eTA;
             thread[8].typ = eT_streamof(intvc10);
   30
             thread[8].obj = (ePtr)objoffset(ta2, heap);
= (ePtr) 10;
             ta2->tvar
Ξ
                          = (ePtr)0;
             ta2->pred
= t1->pTMD;
             ta2->tmd
             ta2->openp
                           = 0;
   35
             /* VAR s */
thread[9].opc = eVAR;
             thread[9].typ = (eType)c10intsml;
H
   40
             /* VAR t */
             thread[10].opc = eVAR;
             thread[10].typ = (eType)intvc10;
             /* EXIT */
   45
             thread[11].opc = eEXIT;
             thread[11].typ = 0;
        #endif
             /* Query 2:
   50
              select x, b
                from S, T
               where y = a
   55
              Plan:
                       BNO -- [ s.#0, t. #1 ]
```

```
NLJ -- [ s.#1 == t.# 0 ]
                     / \
    5
                         ITA T t -- [ t.\#0 > s.\#1 ]
              TA S s
             */
        #if 0
   10
             /* BNO */
             thread[0].opc = eBNO;
             thread[0].typ = eT_streamof(c10vc10);
             thread[0].obj = (ePtr)objoffset(bno, heap);
   15
             bno->in
                          = (ePtr)3;
             bno->fn
                           = (ePtr)1;
             bno->fns
                          = 2;
                          = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
   20
             /* s.#0 */
             thread[1].opc = eVARATT;
             thread[1].typ = eT_setlen(eT_char, 10);
             thread[1].obj = (ePtr)objoffset(varatt1, heap);
varatt1~>tvar = (ePtr)9;
   25
             varatt1->col = 0;
             /* t.#1 */
             thread[2].opc = eVARATT;
thread[2].typ = eT_setlen(eT_varchar, 10);
             thread[2].obj = (ePtr)objoffset(varatt2, heap);
  30
             varatt2->tvar = (ePtr)10;
Harman Harman
             varatt2->col = 1;
             /* NLJ */
   35
             thread[3].opc = eNLJ;
L
             thread[3].typ = 0;
thread[3].obj = (ePtr)objoffset(nlj, heap);
= (ePtr)7;
             nlj->outer
             nlj->inner
                           = (ePtr)8;
   40
             nlj->pred
                          = (ePtr)11;
             /* < */
             thread[4].opc = eLT_II;
             thread[4].typ = eT_bool;
   45
             thread[4].obj = (ePtr)objoffset(lt_ii, heap);
             lt_ii->x1
                        = (ePtr)6;
             lt_ii->x2
                           = (ePtr)5;
             /* t.#0 */
   50
             thread[5].opc = eVARATT;
             thread[5].typ = eT_int;
             thread[5].obj = (ePtr)objoffset(varatt4, heap);
             varatt4->tvar = (ePtr)10;
             varatt4->col = 0;
   55
             /* s.#1 */
             thread[6].opc = eVARATT;
```

```
thread[6].typ = eT_int;
             thread[6].obj = (ePtr)objoffset(varatt3, heap);
             varatt3->tvar = (ePtr)9;
             varatt3->col = 1;
    5
             /* TA S s */
             thread[7].opc = eTA;
             thread[7].typ = eT_streamof(c10intsml);
             thread[7].obj = (ePtr)objoffset(ta1, heap);
   10
                           = (ePtr)9;
             ta1->tvar
             ta1->pred
                           = (ePtr)0;
                           = t-pTMD;
             ta1->tmd
             ta1->openp
                           = 0;
 15
             /* ITA T t */
             thread[8].opc = eITA;
             thread[8].typ = eT_streamof(intvc10);
             thread[8].obj = (ePtr)objoffset(ita, heap);
             ita->tvar
                           = (ePtr)10;
   20
             ita->pred
                           = (ePtr)4;
             ita->tmd
                           = t1->pTMD;
             ita->openp
                           = 0;
<u>ļ.</u>
             ita->ix
                           = *(t1-ppQIscan);
ita->ixid
                           = t1->idxSelected;
   25
             /* VAR s */
             thread[9].opc = eVAR;
             thread[9].typ = (eType)c10intsml;
n
   30
             /* VAR t */
             thread[10].opc = eVAR;
=
thread[10].typ = (eType)intvc10;
             /* == */
   35
             thread[11].opc = eEQ_II;
             thread[11].typ = eT_bool;
thread[11].obj = (ePtr)objoffset(eq_ii, heap);
T.J
                        = (ePtr)5;
             eq_ii->x1
             eq_ii->x2
                           = (ePtr)6;
   40
             /* EXIT */
             thread[12].opc = eEXIT;
             thread[12].typ = 0;
        #endif
   45
             /* Query 3:
                select
                  from S
   50
              order by
              Plan:
                      BNO -- [ * ]
   55
                     SORT s
```

```
TA S
   5
        #if 0
             /* BNO * */
             thread[0].opc = eBNO;
             thread[0].typ = (eType)c10intsml;
   10
             thread[0].obj = (ePtr)objoffset(bno, heap);
             bno->in
                           = (ePtr)1;
             bno->fn
                           = (ePtr)0;
             bno->fns
                           = 3;
                           = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
   15
             /* SORT s */
             thread[1].opc = eSORT;
             thread[1].typ = eT_streamof(c10intsml);
             thread[1].obj = (ePtr)objoffset(sort, heap);
   20
             sort->in
                           = (ePtr)2;
             sort->tvar
                           = (ePtr)0;
                           = query->resultset.pTMD;
             sort->tmd
ļ.
             oStrcpy(sort->tmd->tname, "sorting");
             sort->sortp = 0;
  25
/* TA S */
             thread[2].opc = eTA;
             thread[2].typ = eT_streamof(c10intsml);
             thread[2].obj = (ePtr)objoffset(ta1, heap);
   30
                         = (ePtr)0;
             ta1->tvar
ũ
                           = (ePtr)0;
             ta1->pred
Ξ
             ta1->tmd
                           = t->pTMD;
ta1->openp
                           = 0;
   35
             /* EXIT */
ļ...i.
             thread[3].opc = eEXIT;
thread[3].typ = 0;
        #endif
   40
             /* Query 4:
                select sum(y),count(*)
                  from S
              Plan:
   45
                     BNOAGG -- [ SUM_I,
                                          CNT ]
                                s.#1
   50
                     TA S s
             * /
        #if O
             /* BNOAGG */
   55
             thread[0].opc = eBNOAGG;
             thread[0].typ = (eType)intint;
             thread[0].obj = (ePtr)objoffset(bno, heap);
                                               31
```

```
bno->in
                            = (ePtr) 4;
              bno->fn
                             = (ePtr)1;
              bno->fns
                             = 2;
              bno->bno
                            = oMemAlloc(bno->fns * sizeof(ColVal));
    5
              /* SUM_I */
              thread[1].opc = eSUM_I;
              thread[1].typ = eT_int;
              thread[1].obj = (ePtr)objoffset(sum, heap);
                           = (ePtr)3;
   10
              sum->in
              /* CNT */
              thread[2].opc = eCNT;
              thread[2].typ = eT_int;
   15
              /* s.#1 */
              thread[3].opc = eVARATT;
              thread[3].typ = eT_int;
              thread[3].obj = (ePtr)objoffset(varatt1, heap);
   20
              varatt1->tvar = (ePtr)5;
              varatt1->col = 1;
į...b
              /* TA S */
To the first test that the first test
              thread[4].opc = eTA;
   25
              thread[4].typ = eT_streamof(c10intsml);
              thread[4].obj = (ePtr)objoffset(ta1, heap);
              ta1->tvar
                             = (ePtr)5;
              ta1->pred
                             = (ePtr)0;
              ta1->tmd
                            = t->pTMD;
   30
              ta1->openp
                            = 0;
22
              /* VAR s */
thread[5].opc = eVAR;
              thread[5].typ = (eType)c10intsml;
35
              /* EXIT */
             thread[5].opc = eEXIT;
             thread[5].typ = 0;
        #endif
   40
             /* Query 5:
                      (select (sum(a)
              select
                          from T
   45
                       where a > 1)
                 from
              Plan:
   50
                       BNO -- ATT #0
                             SUBQ
                            AGG -- SUM_I
   55
                                     t.#0
                      TA S
```

```
TA T t -- [t.#0 > 1]
             */
        #if 0
    5
             /* BNO */
             thread[0].opc = eBNO;
             thread[0].typ = (eType)xint;
             thread[0].obj = (ePtr)objoffset(bno, heap);
                          = (ePtr)2;
             bno->in
   10
             bno->fn
                           = (ePtr)1;
                          = 1;
             bno->fns
                           = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
             /* ATT #0 */
   15
             thread[1].opc = eATT;
             thread[1].typ = eT_int;
             thread[1].obj = (ePtr)objoffset(att, heap);
             att->in
                       = (ePtr)3;
             att->col
                          = 0;
   20
             /* TA S */
             thread[2].opc = eTA;
thread[2].typ = eT_streamof(c10intsml);
             thread[2].obj = (ePtr)objoffset(ta1, heap);
   25
             ta1->tvar
                        = (ePtr)0;
             ta1->pred
                          = (ePtr)0;
             ta1->tmd
                          = t - pTMD;
             ta1->openp
                           = 0;
  30
             /* SUBQ */
thread[3].opc = eSUBQ;
Ħ
             thread[3].typ = eT_streamof(xint);
thread[3].obj = (ePtr)objoffset(subq, heap);
             subq->in
                         = (ePtr)4;
   35
subq->tvar
                          = (ePtr)0;
/* AGG */
             thread[4].opc = eAGG;
             thread[4].typ = eT_streamof(xint);
   40
             thread[4].obj = (ePtr)objoffset(agg, heap);
             agg->in
                        = (ePtr)5;
             agg->fn
                          = (ePtr)6;
             agg->fns
                          = 1;
   45
             /* TA T t */
             thread[5].opc = eTA;
             thread[5].typ = eT_streamof(intvc10);
             thread[5].obj = (ePtr)objoffset(ta2, heap);
             ta2->tvar
                          = (ePtr) 10;
   50
             ta2->pred
                          = (ePtr)8;
             ta2->tmd
                          = t1->pTMD;
             ta2->openp
                          = 0;
             /* SUM_I */
  55
            thread[6].opc = eSUM_I;
             thread[6].typ = eT_int;
             thread[6].obj = (ePtr)objoffset(sum, heap);
```

```
sum->in
                          = (ePtr)7;
             /* t.#0 */
             thread[7].opc = eVARATT;
    5
             thread[7].typ = eT_int;
             thread[7].obj = (ePtr)objoffset(varatt1, heap);
             varatt1->tvar = (ePtr)10;
             varatt1->col = 0;
   10
             /* < */
             thread[8].opc = eLT_II;
             thread[8].typ = eT_bool;
             thread[8].obj = (ePtr)objoffset(lt_ii, heap);
             lt_ii->x1
                       = (ePtr)9;
   15
                          = (ePtr)7;
             lt_ii->x2
             /* 1 */
             thread[9].opc = eLIT;
             thread[9].typ = eT_int;
   20
             eT_as_int(thread[9].val) = 1;
             /* VAR t */
Į"į
             thread[10].opc = eVAR;
thread[10].typ = (eType)intvc10;
   25
             /* EXIT */
             thread[11].opc = eEXIT;
             thread[11].typ = 0;
        #endif
   30
             /* Query 6:
₩
select count(z),x,y
                from S
   35
              group by
                          х,у
              Plan:
   40
                          BNO *
                      GROUP +-- [ CNT ]
   45
                           +-- [ MOV, VARATT s.#1 ]
                      SORT as s VARATT s.#0
   50
                    TA S
             */
        #if 0
   55
             /* BNO * */
             thread[0].opc = eBNO;
             thread[0].typ = (eType)intc10int;
                                               34
```

```
thread[0].obj = (ePtr)objoffset(bno, heap);
         bno->in
                       = (ePtr)1;
         bno->fn
                       = (ePtr)0;
         bno->fns
                       = 3;
5
         bno->bno
                       = oMemAlloc(bno->fns * sizeof(ColVal));
          /* GROUP */
         thread[1].opc = eGROUP;
         thread[1].typ = eT_streamof(c10intint);
         thread[1].obj = (ePtr)objoffset(group, heap);
10
                       = (ePtr)5;
         group->in
                        = (ePtr)5;
         group->src
                        = (ePtr)2;
         group->agg
                       = 1;
         group->aggs
         group->fns
15
                       = 2;
          /* CNT */
          thread[2].opc = eCNT;
          thread[2].typ = eT_int;
20
          /* MOV */
         thread[3].opc = eMOV;
          thread[3].typ = eT_setlen(eT_char, 10);
          thread[3].obj = (ePtr)objoffset(mov, heap);
25
                      = (ePtr)8;
         mov->in
                        = 10+1;
         mov->n
          /* VARATT s.#1 */
          thread[4].opc = eVARATT;
          thread[4].typ = eT_int;
30
          thread[4].obj = (ePtr)objoffset(varatt2, heap);
         varatt2->tvar = (ePtr)7;
          varatt2->col = 1;
35
          /* SORT */
          thread[5].opc = eSORT;
          thread[5].typ = eT_streamof(c10intsml);
          thread[5].obj = (ePtr)objoffset(sort, heap);
          sort->in
                        = (ePtr)6;
          sort->tvar
40
                        = (ePtr)7;
                        = gb->table->pTMD;
          sort->tmd
          oStrcpy(sort->tmd->tname, "teggy11");
          sort->sortp
                       = 0;
          /* TA S */
45
          thread[6].opc = eTA;
          thread[6].typ = eT_streamof(c10intsml);
          thread[6].obj = (ePtr)objoffset(ta1, heap);
                        = (ePtr)0;
          ta1->tvar
50
          ta1->pred
                        = (ePtr)0;
          tal->tmd
                        = t->pTMD;
                        = 0;
          ta1->openp
          /* VAR s */
          thread[7].opc = eVAR;
55
          thread[7].typ = (eType)c10intsml;
```

n.

```
/* VARATT s.#0 */
             thread[8].opc = eVARATT;
             thread[8].typ = eT_setlen(eT_char, 10);
             thread[8].obj = (ePtr)objoffset(varatt1, heap);
   5
            varatt1->tvar = (ePtr)7;
            varatt1->col = 0;
             /* EXIT */
             thread[9].opc = eEXIT;
   10
             thread[9].typ = 0;
        #endif
        #if 0
             /* BNO * */
   15
             thread[0].opc = eBNO;
             thread[0].typ = (eType)intd105int;
             thread[0].obj = (ePtr)objoffset(bno, heap);
                           = (ePtr)1;
             bno->in
   20
             bno->fn
                           = (ePtr)0;
             bno->fns
                           = 3;
                           = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
Į.i.
E STATE
             /* GROUP */
25
             thread[1].opc = eGROUP;
             thread[1].typ = eT_streamof(d105intint);
             thread[1].obj = (ePtr)objoffset(group, heap);
                           = (ePtr)5;
             group->in
                           = (ePtr)5;
             group->src
                           = (ePtr)2;
   30
             group->agg
             group->aggs
                           = 1;
50
             group->fns
                           = 2;
/* CNT */
   35
             thread[2].opc = eCNT;
ţ.i.l
             thread[2].typ = eT_int;
/* VARATT s.#0 */
N
             thread[3].opc = eMOV;
             thread[3].typ = eT_setscale(eT_setprec(eT_decimal, 10), 5);
   40
             thread[3].obj = (ePtr)objoffset(mov, heap);
                           = (ePtr)8;
             mov->in
                           = MAX_INTERNAL_DECIMAL_BYTES;
             mov->n
             /* VARATT s.#1 */
   45
             thread[4].opc = eVARATT;
             thread[4].typ = eT_int;
             thread[4].obj = (ePtr)objoffset(varatt2, heap);
             varatt2->tvar = (ePtr)7;
             varatt2->col = 1;
   50
             /* SORT */
             thread[5].opc = eSORT;
             thread[5].typ = eT_streamof(d105intsml);
             thread[5].obj = (ePtr)objoffset(sort, heap);
   55
                           = (ePtr)6;
             sort->in
                            = (ePtr)7;
             sort->tvar
```

```
sort->tmd
                           = gb->table->pTMD;
             oStrcpy(sort->tmd->tname, "teggy10");
             sort->sortp
                          = 0;
             /* TA S */
    5
             thread[6].opc = eTA;
             thread[6].typ = eT_streamof(d105intsml);
             thread[6].obj = (ePtr)objoffset(ta1, heap);
             ta1->tvar
                           = (ePtr)0;
   10
             ta1->pred
                           = (ePtr)0;
             ta1->tmd
                           = t-pTMD;
                           = 0;
             ta1->openp
             /* VAR s */
   15
             thread[7].opc = eVAR;
             thread[7].typ = (eType)d105intsml;
             /* VARATT s.#0 */
             thread[8].opc = eVARATT;
             thread[8].typ = eT_setscale(eT_setprec(eT_decimal, 10), 5);
   20
             thread[8].obj = (ePtr)objoffset(varatt1, heap);
             varatt1->tvar = (ePtr)7;
             varatt1->col = 0;
25
             /* EXIT */
             thread[9].opc = eEXIT;
             thread[9].typ = 0;
        #endif
   30
             /* Query 7:
3
select x + y
                from S
   35
Plan:
                    BNO ---- [ ADD_DI ]
   40
                 TA S s
                         VARATT s.#1
                                        VARATT s.#2
             * /
        #if O
   45
             /* BNO */
             thread[0].opc = eBNO;
             thread[0].typ = (eType)d105;
             thread[0].obj = (ePtr)objoffset(bno, heap);
             bno->in
                           = (ePtr)4;
   50
             bno->fn
                           = (ePtr)1;
             bno->fns
                           = 1;
                           = oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
             /* ADD DI */
   55
             thread[1].opc = eADD_DI;
             thread[1].typ = eT_setscale(eT_setprec(eT_decimal, 10), 5);
             thread[1].obj = (ePtr)objoffset(add_di, heap);
```

```
add_di -> x1 = (ePtr)2;
             add di -> x2 = (ePtr)3;
             /* VARATT */
    5
             thread[2].opc = eVARATT;
             thread[2].typ = eT_setscale(eT_setprec(eT_decimal, 10), 5);
             thread[2].obj = (ePtr)objoffset(varatt1, heap);
             varatt1->tvar = (ePtr)5;
             varatt1->col = 1;
   10
             /* VARATT */
             thread[3].opc = eVARATT;
             thread[3].typ = eT_int;
             thread[3].obj = (ePtr)objoffset(varatt2, heap);
   15
             varatt2->tvar = (ePtr)5;
             varatt2->col = 2;
             /* TA S */
             thread[4].opc = eTA;
   20
             thread[4].typ = eT_streamof(intd105int);
             thread[4].obj = (ePtr)objoffset(ta1, heap);
             ta1->tvar = (ePtr)5;
                         = (ePtr)0;
             ta1->pred
             ta1->tmd
                         = t-pTMD;
   25
             ta1->openp
                          = 0;
/* VAR s */
thread[5].opc = eVAR;
             thread[5].typ = (eType)intd105int;
m
   30
CI.
             /* EXIT */
æ
             thread[6].opc = eEXIT;
thread[6].typ = 0;
   35
        #endif
/* Query 8:
              select y - z
   40
                from S
                Plan:
                    BNO ---- [ SUB_II ]
   45
                 TA S s VARATT s.#1 VARATT s.#2
   50
             /* BNO */
             thread[0].opc = eBNO;
             thread[0].typ = (eType)xint;
             thread[0].obj = (ePtr)objoffset(bno, heap);
                         = (ePtr)4;
   55
             bno->in
             bno->fn
                          = (ePtr)1;
                          = 1;
             bno->fns
                                              38
```

```
= oMemAlloc(bno->fns * sizeof(ColVal));
             bno->bno
             /* SUB_II */
             thread[1].opc = eSUB_II;
   5
             thread[1].typ = eT_int;
             thread[1].obj = (ePtr)objoffset(sub_ii, heap);
             sub_ii->x1
                          = (ePtr)2;
             sub_ii->x2
                           = (ePtr)3;
             /* VARATT */
   10
             thread[2].opc = eVARATT;
             thread[2].typ = eT_int;
             thread[2].obj = (ePtr)objoffset(varatt1, heap);
             varatt1->tvar = (ePtr)5;
   15
             varatt1->col = 1;
             /* VARATT */
             thread[3].opc = eVARATT;
             thread[3].typ = eT_smlint;
             thread[3].obj = (ePtr)objoffset(varatt2, heap);
   20
             varatt2->tvar = (ePtr)5;
             varatt2->col = 2;
<u>L</u>L
/* TA S */
             thread[4].opc = eTA;
   25
             thread[4].typ = eT_streamof(c10intsml);
             thread[4].obj = (ePtr)objoffset(ta1, heap);
             ta1->tvar
                            = (ePtr)5;
                           = (ePtr)0;
             ta1->pred
m
   30
             ta1->tmd
                           = t->pTMD;
ū
                            = 0;
             ta1->openp
31
/* VAR s */
IJ
             thread[5].opc = eVAR;
   35
             thread[5].typ = (eType)c10intsml;
į.
Hall Hand
             /* EXIT */
             thread[6].opc = eEXIT;
             thread[6].typ = 0;
   40
             ePrepare(&s);
             while ((rc = eInterpret(&s)) == eOK) {
   45
                eRender(&s, hostio);
             }
             eSweep(&s);
   50
             oMemFreeSafe(bno->bno);
        }
```